NEPI: Network Experimentation Programing Interface

Alina Quereilhac, Thierry Turletti, Walid Dabbous[†]

http://nepi.inria.fr

[†] the authors are not liable for any mistake the presenter will make

How can we make it really simple to run ICN experiments in the wild?

Experiments issues

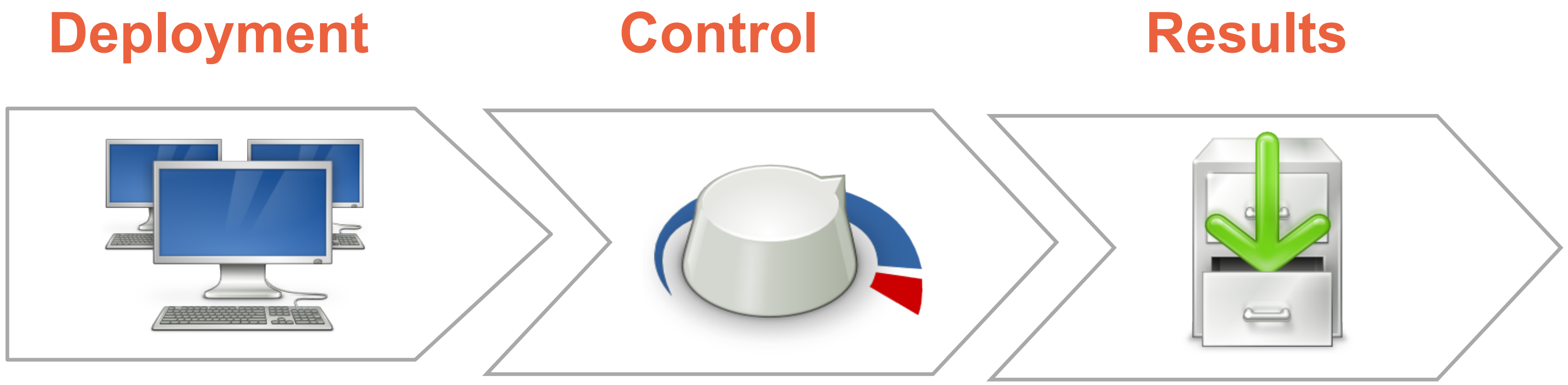
- Once you master the testbed you still have to
  - implement the experiment,
  - synchronize the resource needed for the experiment,
  - detect and handle errors during execution,
  - collect results.
- Automation alleviates these issues.

NEPI: Network Experiment Programming Interface

NEPI in a nutshell

- NEPI is a framework to manage network experiments
  - that abstracts testbed differences behind a common interface
  - to automate experimentation steps.
- NEPI runs on the user side (e.g., user desktop)
  - i.e., no need to modify the testbed.

Experiment automation



- Configuration resource
- Install software
- Synchronization start
- Instrument resource
- Start resource

- Changes configuration
- Monitor status
- Detect errors
- Release resources

- Query information
- Download results

Everything is a resource

- The user interacts with the Experiment Controller (EC), which controls the Resource Managers.
- The Resource Managers (RMs) control individual resources (1 RM per resource type)
- All RMs implement a same interface
  - e.g., `deploy`, `start`, `stop`.
- An experiment is a graph of interconnected resources.

A ping example

```
from nepi.execution.ec import ExperimentController
ec = ExperimentController()

node = ec.register_resource("LinuxNode")
ec.set(node, "hostname", “planetlab1.inria.fr”)
ec.set(node, "username", “me”)

app = ec.register_resource("LinuxApplication")
ec.set(app, "command", "ping -c3 nepi.inria.fr")
ec.register_connection(app, node)

ec.deploy()

ec.wait_finished(app)

ec.shutdown()
```

Ongoing work

- We (with Priya) asses the costs/benefits of CCN overlays by deploying CCNx on PlanetLab
  - impact of topologies?
  - impact of CCN parameters?
  - impact of traffic patterns?

NEPI status

- Supported testbeds:
  - (any) Linux host with SSH key authentication,
  - PlanetLab testbed,
  - OMF wireless testbeds (under test).
- Other testbeds:
  - Amazon EC (should work. untested), Grid5000 (should work. untested), ns-3 (ongoing).
- Virtually any other testbed (= set of resources).

Trying out NEPI?

- NEPI is implemented in Python.
- NEPI 3.0 to be released soon (with documentation and examples)
  - web http://nepi.inria.fr,
  - mailing list: nepi-users@inria.fr,
    - send an email to sympa@inria.fr with subject *SUBscribe nepi-users* <*your-username*>.

NEPI: Network Experimentation Programing Interface

Alina Quereilhac, Thierry Turletti, Walid Dabbous[†]

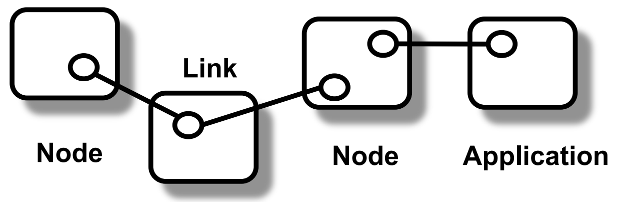
http://nepi.inria.fr

[†] the authors are not liable for any mistake the presenter made

Complementary material

Experiment representation

- Experiments are represented as a graph of interconnected resources.



- Each resources have 3 set of properties:
  - attributes (e.g., configuration)
  - traces (e.g., stderr, stdout)
  - states (i.e., STARTED, STOPPED, FAILED)

A CCNx example on PlanetLab

```
from nepi.execution.ec import ExperimentController
ec = ExperimentController()
node = ec.register_resource("LinuxNode")
ec.set(node, "hostname", “planetlab1.inria.fr”)
ec.set(node, "username", “me”)

ccnd = ec.register_resource("LinuxCCND")
ec.register_connection(ccnd, node)

ccnr = ec.register_resource("LinuxCCNR")
ec.register_connection(ccnr, ccnd)

entry = ec.register_resource("LinuxFIBEntry")
ec.set(entry, "host", “planetlab2.usa.org”)
ec.register_connection(entry, ccnd)
```